10/16/2019

# ISM 6218: Advanced Database

EXAM - 2

Ezema, Chukwuebuka
TEAM ENJOY

# Table of Contents

1. **Create 3 User Defined Functions, using all the following:**

    a) **Union, Union ALL, Coalesce, is null, exist, having, IN, Any**
    b) **Except and Intersect operators**
    c) **1 must return a table, 2 must return a single value**
    d) **Call 1 UDF from a SPROC**
    e) **Call 1 UDF from Cursor**
    f) **Call 1 UDF from CTE**
    g) **Use Apply operator to join UDF with another table**

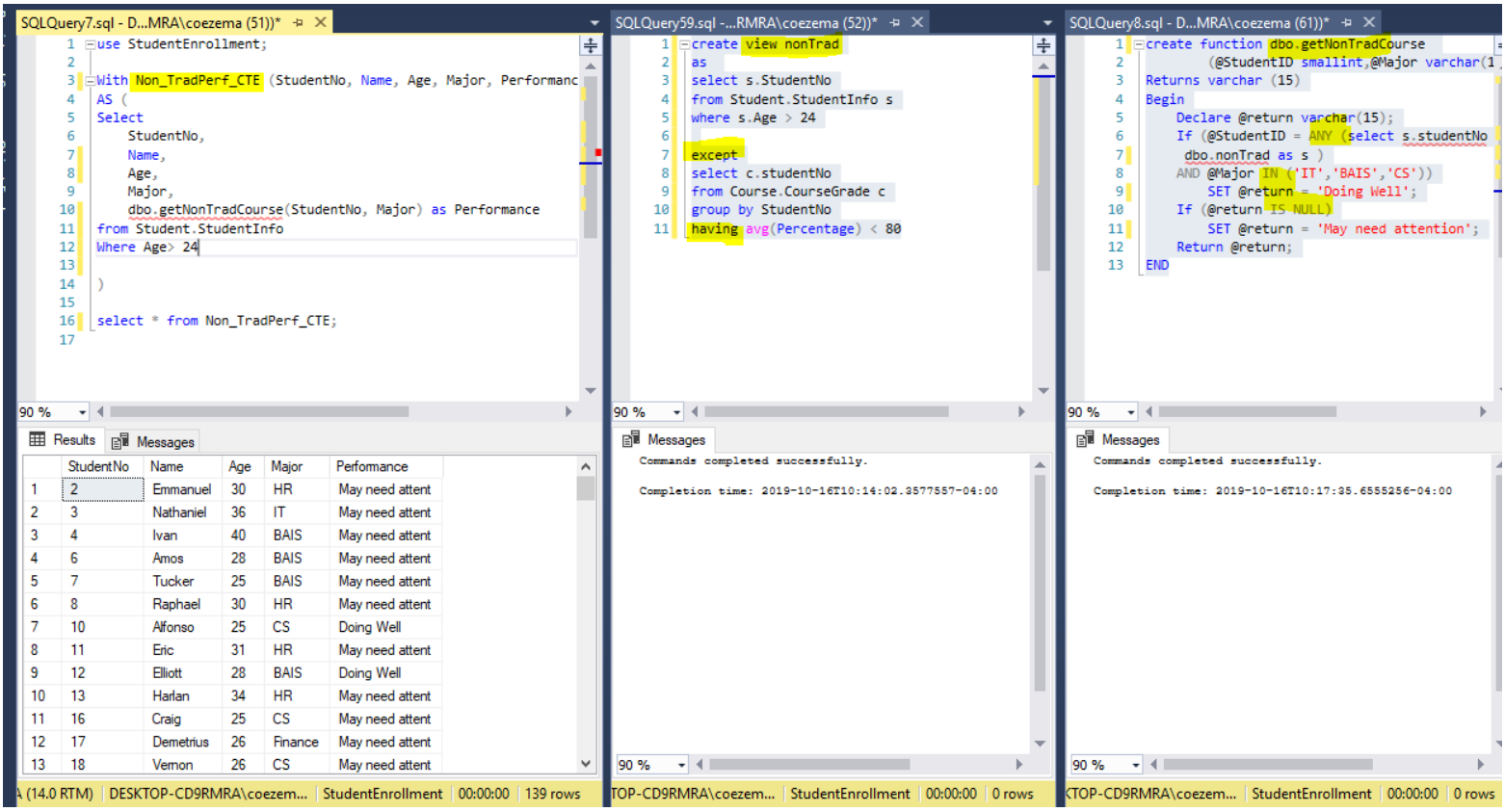
## User Story 1:

As the BAIS Academic Advisor, I want a report of all the non traditional students with their current GPA, so that I can evaluate it against the class average.

Requirements for user story:

- Create a view of all the non-traditional students Student who are older than 24.
- Get course average
- Check the performance using a scalar UDF.
- Fetch all the student performance info using a CTE.

**Used: CTE along with scalar UDF and Operators (IS NULL, Having, IN, ANY, EXCEPT)**



**Results:**

From the CTE results, we can see that:

- There are quite a few students that may need our attention, this may be due to some of our students just starting in the program.

## User Story 2:

As the academic advisor I want a report of the non traditional students from Human resource and Finance department who failed their exam so that I can contact them about possible academic probation.

Requirements for user story:

- Get HR and Finance non traditional student data
- Get contact details(email/phone) for each such student.

**Used: SPROC along with table-valued UDF and operators (Coalesce, UNION, UNION ALL) and Cross Apply**

SQLQuery10.sql -...RMRA\coezema (54))*

```sql
CREATE FUNCTION getPoorNTradStudents (@StudentNo varchar (50))
Returns TABLE
AS RETURN
SELECT * FROM Course.CourseGrade AS GRADE
        WHERE GRADE.StudentNo = @StudentNo
            AND GRADE.Grade = 'F';

CREATE PROC GetStudentsContact
As  SELECT S.StudentNo, S.Name, COALESCE (S.Email, S.PhoneNo) as 'Contact Info', s.Major, grade
    FROM Student.StudentInfo AS S Cross Apply getPoorNTradStudents (S.StudentNo) as Grade
    where s.Major = 'HR'
    and s.Age>24
    UNION
    SELECT S.StudentNo, S.Name, COALESCE (S.Email, S.PhoneNo) as 'Contact Info', s.Major, grade
    FROM Student.StudentInfo AS S Cross Apply getPoorNTradStudents (S.StudentNo) as Grade
    where s.Major = 'Finance'
    and s.Age>24
    Order BY S.StudentNo
    go
exec GetStudentsContact
```

90 %

Results | Messages

| | StudentNo | Name | Contact Info | Major | grade |
|---|---|---|---|---|---|
| 1 | 8 | Raphael | 890163507 | HR | F |
| 2 | 11 | Eric | uma.Nunc@facilisisSuspendissecommodo.co.uk | HR | F |
| 3 | 17 | Demetrius | mauris.a.nunc@famesacturpis.net | Finance | F |
| 4 | 24 | Magee | 833922965 | Finance | F |
| 5 | 39 | Jermaine | faucibus.leo@eros.co.uk | Finance | F |
| 6 | 48 | William | ullamcorper@nisiMauris.co.uk | HR | F |
| 7 | 64 | Tate | Proin@posuereenim.org | HR | F |
| 8 | 84 | Zephania | lacus@faucibuslectusa.co.uk | HR | F |
| 9 | 90 | Seth | tristique.neque@Nuncullamcorper.ca | HR | F |
| 10 | 101 | Lila | consequat.enim@vulputatelacus.org | HR | F |
| 11 | 105 | Alea | et.magnis@mollisPhaselluslibero.co.uk | Finance | F |
| 12 | 109 | Iliana | Integer.vitae@congue.net | Finance | F |
| 13 | 125 | Zelda | semper.pretium.neque@acmattisvelit.net | Finance | F |

Click to select the whole column

Query execu... | DESKTOP-CD9RMRA (14.0 RTM) | DESKTOP-CD9RMRA\coezem... | StudentEnrollment | 00:00:00 | 15 rows

SQLQuery63.sql -...RMRA\coezema (51))*

```sql
CREATE PROC GetStudentsContactALL
As  SELECT S.StudentNo, S.Name, COALESCE (S.Email, S.PhoneNo) as 'Cont
    FROM Student.StudentInfo AS S Cross Apply getPoorNTradStudents (S.
    where s.Major = 'HR'
    and s.Age>24
    UNION ALL
    SELECT S.StudentNo, S.Name, COALESCE (S.Email, S.PhoneNo) as 'Cont
    FROM Student.StudentInfo AS S Cross Apply getPoorNTradStudents (S.
    where s.Major = 'Finance'
    and s.Age>24
    Order BY S.StudentNo
    go
exec GetStudentsContactALL
```

90 %

Results | Messages

| | StudentNo | Name | Contact Info | Major | grade |
|---|---|---|---|---|---|
| 1 | 8 | Raphael | 890163507 | HR | F |
| 2 | 11 | Eric | uma.Nunc@facilisisSuspendissecommodo.co.uk | HR | F |
| 3 | 11 | Eric | uma.Nunc@facilisisSuspendissecommodo.co.uk | HR | F |
| 4 | 17 | Demetrius | mauris.a.nunc@famesacturpis.net | Finance | F |
| 5 | 24 | Magee | 833922965 | Finance | F |
| 6 | 39 | Jermaine | faucibus.leo@eros.co.uk | Finance | F |
| 7 | 48 | William | ullamcorper@nisiMauris.co.uk | HR | F |
| 8 | 64 | Tate | Proin@posuereenim.org | HR | F |
| 9 | 84 | Zephania | lacus@faucibuslectusa.co.uk | HR | F |
| 10 | 90 | Seth | tristique.neque@Nuncullamcorper.ca | HR | F |
| 11 | 101 | Lila | consequat.enim@vulputatelacus.org | HR | F |
| 12 | 105 | Alea | et.magnis@mollisPhaselluslibero.co.uk | Finance | F |
| 13 | 105 | Alea | et.magnis@mollisPhaselluslibero.co.uk | Finance | F |

-CD9RMRA (14.0 RTM) | DESKTOP-CD9RMRA\coezem... | StudentEnrollment | 00:00:00 | 17 rows

**Results**:

- Total 15 Non tradition students received an F and are in jeopardy of getting an academic probation.

What happens if we use UNION ALL instead of UNION?

**Results**:

- It returned 17 records.
- Union ALLincluded duplicate records as well.

## User Story 3:

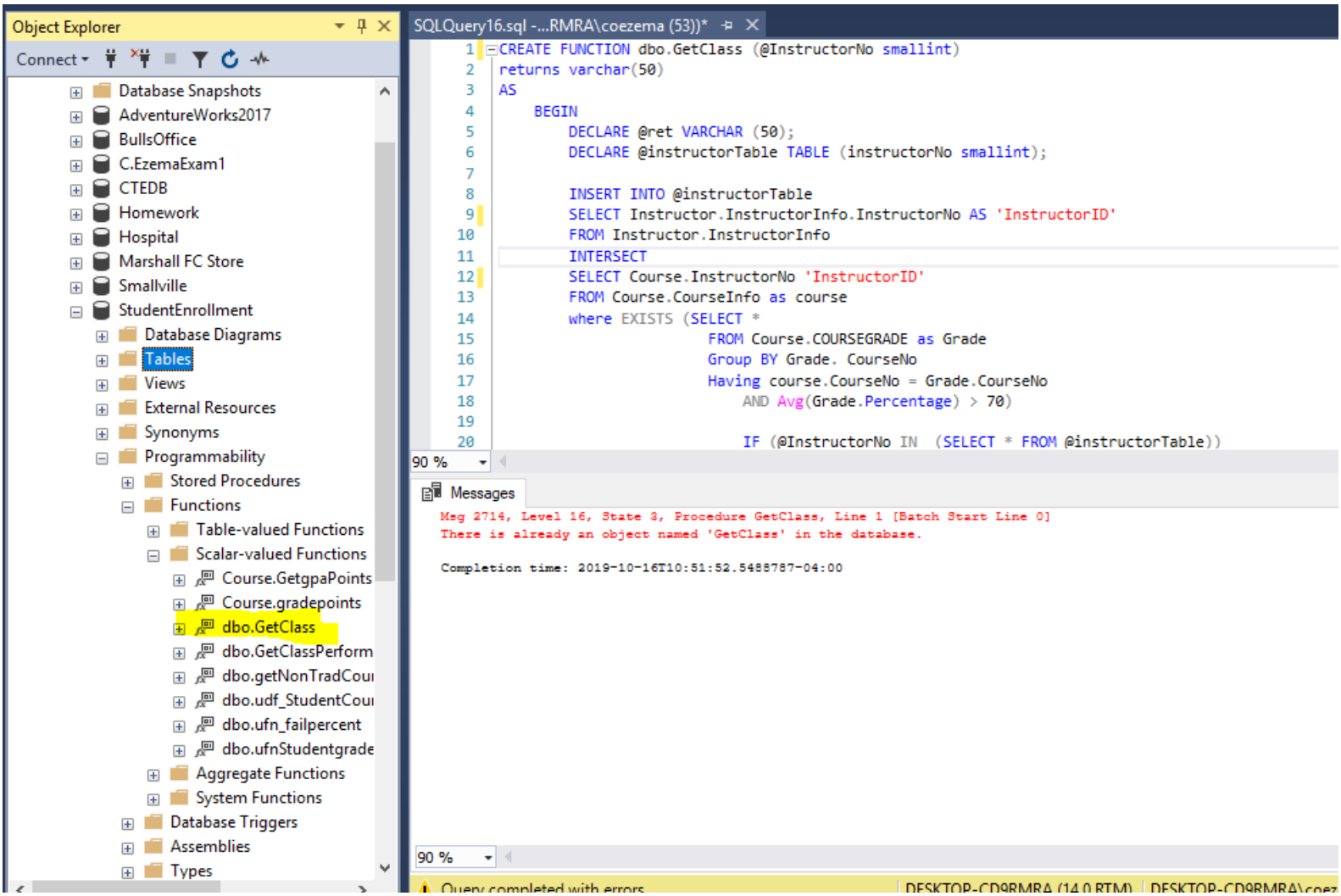
The dean of the university wants a report of every instructor's class performance to evaluate difficulty level and how students are performing in them.

Requirements of the story:

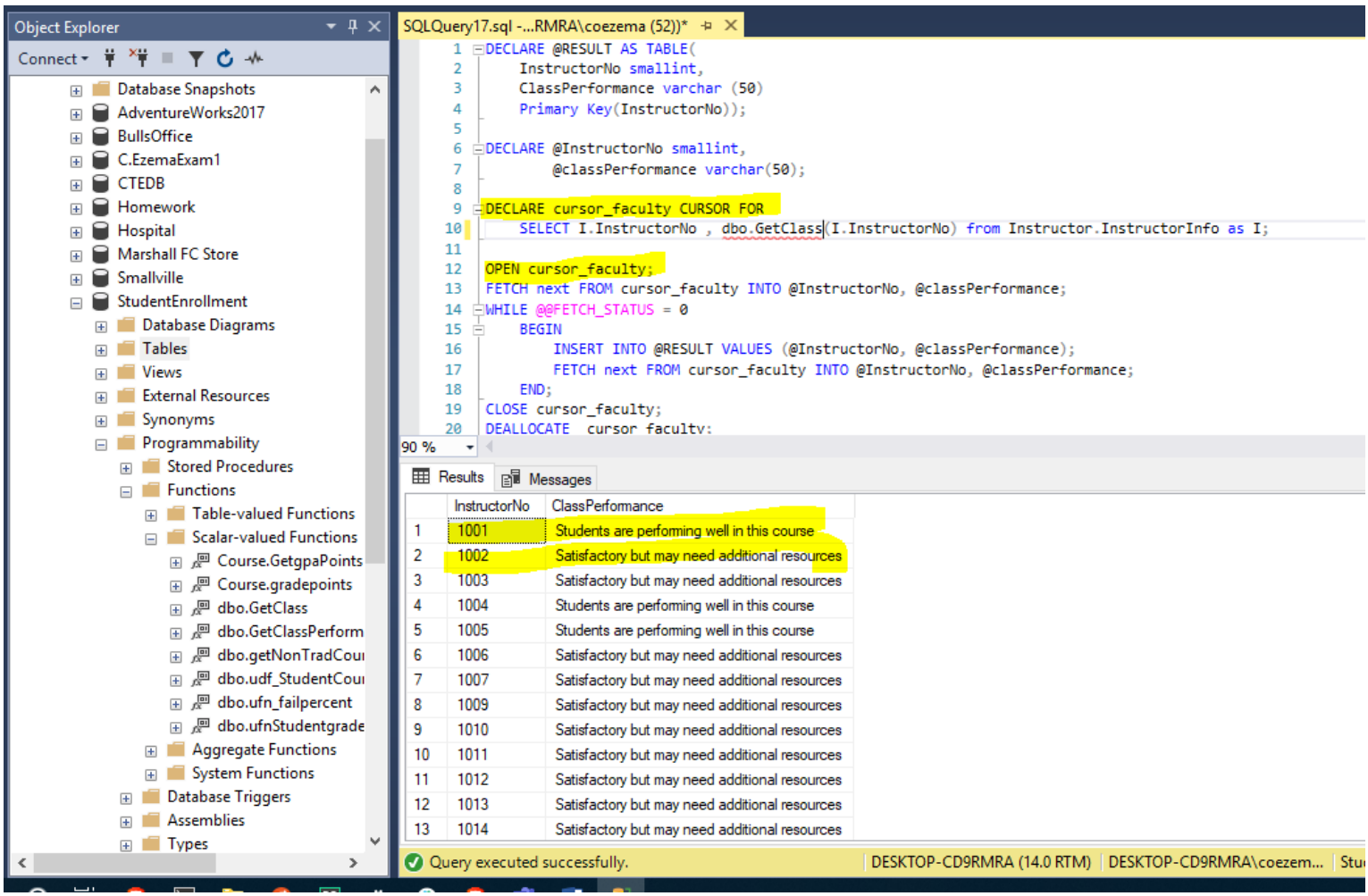
- Create a function to classify the instructors based on their course difficulty- average grade for student
- Call the function from a Cursor.

**Used: Cursor along with scalar UDF and operators (INTERSECT, EXISTS, IN, IS NULL)**

Let's start by creating a scalar UDF which will classify the class performance for every instructor. If the average class performance is above 70%, then course difficulty is deemed appropriate.



Let’s now call the function from a cursor.




**Results**:

- Every instructor’s class performance has been classified as needing additional resources or students are performing well. If the course average is below 70% then we are not getting the nice bell curve that denotes a normal level of difficulty. There seems to be quite a few courses that may need additional resource and further evaluation will be required prior to committing t

2. **Create a Parameterized SQL Query**
    a) **Use Exec SQL**
    b) **Use @SQL**

# User Case Story

As the Program Director, I want to know if each course's difficulty is appropriate for students, so that additional resources can be provided if needed.

STORY REQUIREMENTS:

- Course difficulty is determined by average grade of course
- Course is considered difficult/challenging if mean is below 70
- Course Name should be displayed along with course average
- Difficulty status should be printed

**Using Exec SQL**

SQLQuery52.sql -...RMRA\coezema (57))*

```sql
DECLARE @sql varchar(max), @columnList varchar(75), @classAvg numeric(5,2)
DECLARE @coursePop int
DECLARE @course int
DECLARE @status varchar(255)

SET @course = 10004
SET @classAvg = (SELECT avg(percentage) from Course.CourseGrade WHERE CourseNo = @course)
SET @coursePop = (SELECT count(StudentNo) from Course.CourseGrade WHERE CourseNo = @course)
SET @columnList = 'c.CourseNo,ci.Name,'


IF ((@classAvg) < 70.00)  SET @status =  '''Students find this course challenging'''
ELSE SET @status = '''Difficulty Level is appropriate'''

SET @sql =  'SELECT ' + @columnList + cast(@classAvg as varchar(255))+'as Class'+','+ @status+' Course_Status'+
                    ' FROM Course.CourseGrade c join Course.CourseInfo ci
                      on c.CourseNo = ci.CourseNo
                      where c.CourseNo = ' + cast(@course as varchar(255))+
                      'group by c.CourseNo, ci.Name'
EXEC(@sql)
```

91 %

Results | Messages

| | CourseNo | Name | Class | Course_Status |
|---|---|---|---|---|
| 1 | 10004 | Analytical Methods for Business | 63.82 | Students find this course challenging |

Logic check using course 10005.

90 %

Results | Messages

| | CourseNo | Class Average | Course_Status |
|---|---|---|---|
| 1 | 10005 | 74.31 | Student are able to perform optimally |

**Using @SQL:**

SQLQuery54.sql -...RMRA\coezema (54))* | SQLQuery52.sql -...RMRA\coezema (57))*

```sql
DECLARE @sql nvarchar(max), @columnList varchar(75), @classAvg numeric(5,2)
DECLARE @coursePop int
DECLARE @course int
DECLARE @status varchar(255)

SET @course = 10004
SET @classAvg = (SELECT avg(percentage) from Course.CourseGrade WHERE CourseNo = @course)
SET @coursePop = (SELECT count(StudentNo) from Course.CourseGrade WHERE CourseNo = @course)
SET @columnList = 'c.CourseNo,ci.Name,'


IF ((@classAvg) < 70.00)  SET @status =  '''Students find this course challenging'''
ELSE SET @status = '''Difficulty Level is appropriate'''

SET @sql =  'SELECT ' + @columnList + cast(@classAvg as varchar(255))+'as Class'+','+ @status+' Course_Status'+
                    ' FROM Course.CourseGrade c join Course.CourseInfo ci
                      on c.CourseNo = ci.CourseNo
                      where c.CourseNo = @course
                      group by c.CourseNo, ci.Name'

EXECUTE sp_executesql @sql, N'@course varchar(75)', @course = @course
```

91 %

Results | Messages

| | CourseNo | Name | Class | Course_Status |
|---|---|---|---|---|
| 1 | 10004 | Analytical Methods for Business | 63.82 | Students find this course challenging |

Results:

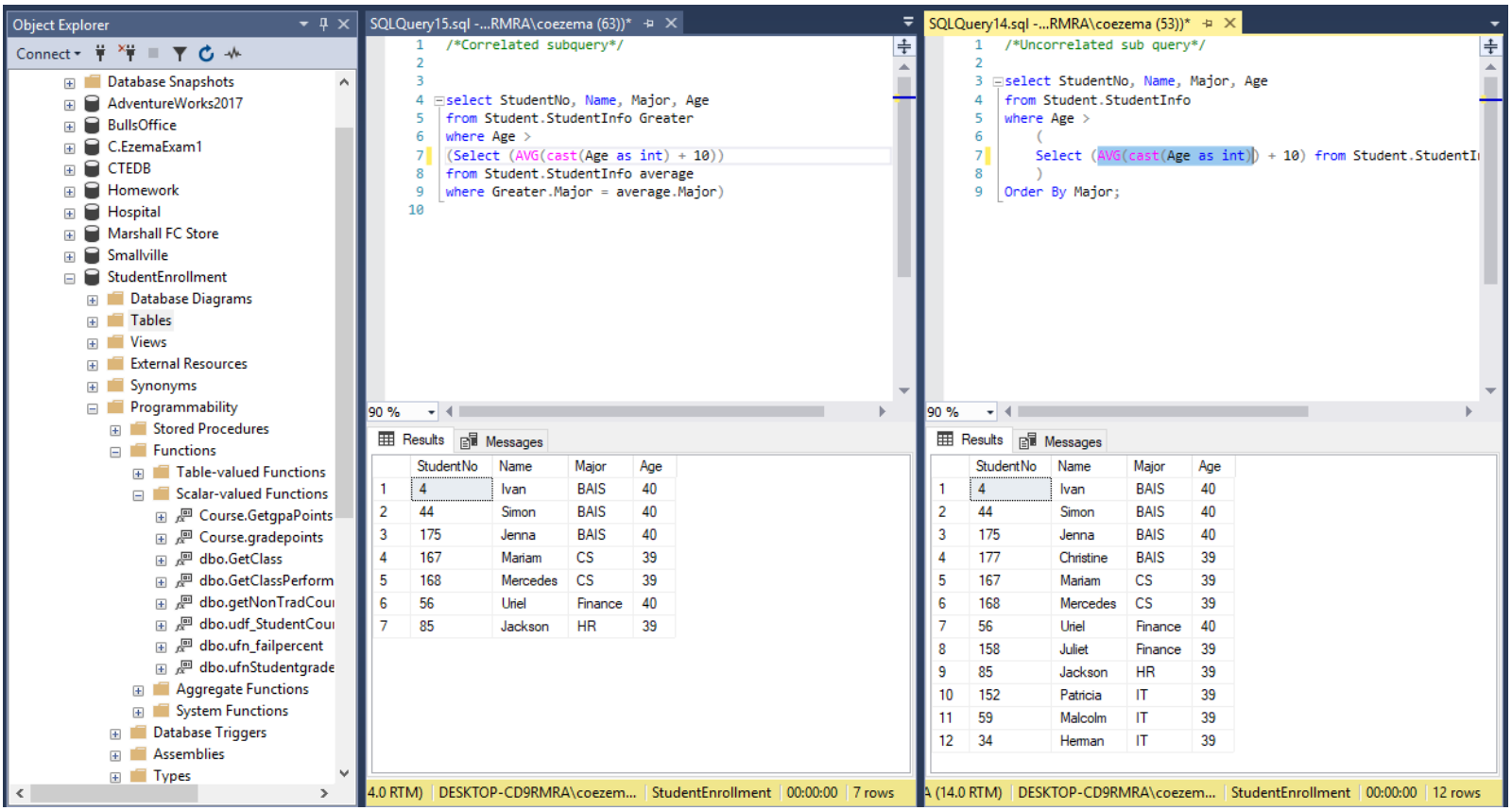
- The Program director can quickly assess the class performance by inserting the course No.  if students are performing poorly on average, then the director can make additional resources like TA ‘s and GA available for students to boost the class average.

### 3. Create 2 Sub queries: 1 correlated, 1 uncorrelated

**User story:** Program director wants a report of all the nontraditional students (age at least 10 years greater than the average in the major/dept) for the USNEWS college population reporting.

- **Correlated Subquery**

**Uncorrelated Subquery**



4. **Create any query using a set of Case Expressions**

**User Story:**

As the college admission committee, I want to know which age_bucket a student belongs so that I can develop dynamic reports with that data set.

User story requirements:

- Insert age bucket column for report developers to use

for case.sql - DES...RMRA\coezema (58))* | SQLQuery72.sql -...RMRA\coezema (73))* | SQLQuery71.sql -...RMI

```sql
SELECT s.studentNo, S.Name as 'Student Name',
        count(c.courseNo) as 'TotalCourses',
        Cast( avg(c.percentage) As Decimal (4,2)) as 'Avg Percentage',
        s.Age,
        CASE
           WHEN Age  >39 THEN 'Bob is your Uncle'
           WHEN Age  > 35 AND Age  <= 39 THEN 'Mid-Career'
           WHEN Age  >= 21 AND Age  <= 35 THEN 'Adult'
           WHEN Age  >= 18 AND Age  <= 20 THEN 'Young Adult'
           ELSE 'Not Found'
        END as Age_Bucket
FROM  Student.StudentInfo s
join Course.CourseGrade c
on s.studentNo = c.StudentNo
group by s.studentNo, s.name,Age
order by s.studentNo
```

90 %

Results | Messages

| | studentNo | Student Name | TotalCourses | Avg Percentage | Age | Age_Bucket |
|---|---|---|---|---|---|---|
| 1 | 1 | Ivor | 4 | 66.00 | 18 | Young Adult |
| 2 | 2 | Emmanuel | 1 | 71.00 | 30 | Adult |
| 3 | 3 | Nathaniel | 3 | 65.67 | 36 | Mid-Career |
| 4 | 4 | Ivan | 2 | 68.50 | 40 | Bob is your Uncle |
| 5 | 5 | Chaim | 2 | 76.50 | 21 | Adult |
| 6 | 6 | Amos | 8 | 71.75 | 28 | Adult |
| 7 | 7 | Tucker | 1 | 60.00 | 25 | Adult |
| 8 | 8 | Raphael | 1 | 48.00 | 30 | Adult |
| 9 | 9 | Bruno | 1 | 66.00 | 19 | Young Adult |
| 10 | 11 | Eric | 3 | 61.33 | 31 | Adult |
| 11 | 13 | Harlan | 2 | 73.50 | 34 | Adult |
| 12 | 14 | Aladdin | 2 | 46.50 | 20 | Young Adult |
| 13 | 15 | Upton | 1 | 41.00 | 23 | Adult |

Query executed successfully.

Results:

- We were able to add the age column for report developers.